GDQA : Graph Driven Question Answering System
– NTCIR-4 QAC2 Experiments –

Gakuto KURATA Naoaki OKAZAKI Mitsuru ISHIZUKA
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{kurata, okazaki, ishizuka}@miv.t.u-tokyo.ac.jp

Abstract

In this paper, we present a question answering system developed for NTCIR-4 in detail. Our question answering system, GDQA, employs a new text retrieval algorithm specialized for QA and a new algorithm for sorting the answer candidates. Using our new algorithm for text retrieval, articles containing the asnwer for the question can be retirieved with high precision. Our algorithm for sorting the answer candidates uses the graph structure based on the result of the dependency analysis of retrieved articles. With this sorting algorithm, GDQA can present the correct answer in a higher rank than the other candidates.

Keywords: *Question Answering, Dependency Analysis, Graph Structure, GETA, QAC*

1 Introduction

Question Answering(QA) is a task to present the appropriate answer for the question written in the natural language from big corpus available in the computerized forms. QA is widely noticed as the complex of Information Retrieval, Information Extraction and other natural language processing techniques. A lot of researchers are dealing with the QA task and the evaluation workshops, such as TREC and NTCIR, designed to enhance research are held.

Tab. 1 shows the difference between QA and IR(Information Retrieval).

Using IR system like Google, we can only get the documents related to the terms we inputted. So, we must read the documents through and look for the information we need. On the other hand, we can directly get the answer for the query by using QA system. QA system saves our labor in the IR system shown in below.

- To determine the terms from our information need in mind
- To read the documents to seek the information we need

Ubiquious computing era is in coming. You can easily imagine the situation that people talk to the computers distributed around and ask some questions and then computers return the answer. To realize this vision, Question Answering and Speech Recgnition will be essential.

According to this vision, QA is the promising field. However, the performance of the current QA system is very poor. In this paper, we present new techniques to improve the performance.

2 Related works

A lot of systems have been estbalished. Most of the systems employ 4 steps. Fig. 1 shows the common architecture of the conventional QA systems.

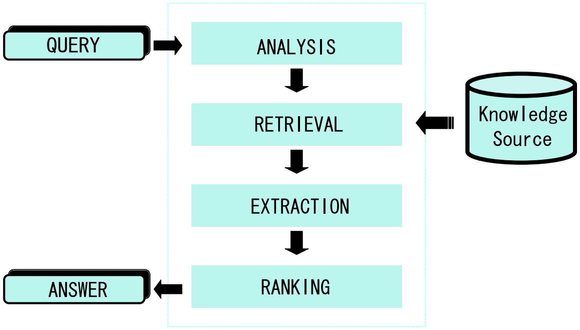

Figure 1. Common Architecture of QA systems

We will briefly describe the appraoch in the conventional QA systems to realize the 4 procedure shown in Fig. 1. At the same time, we will point out the problems of the former approaches.

Table 1. Difference between QA and IR

	QA	IR
Input	Queries written in natural language	Query terms
Output	Answers for the query	Related documents

2.1 Common Architecture of QA system

2.1.1 ANALYSIS of the query

In the first step, queries written in natural language are analyzed.

For example, a query like "Who is the president of U.S.A.?" indicates that the answer is the name of a person. Interrogative words like who, where and when suggests the answer category. Tab. 2 shows the interrogative words in Japanese and categories they suggest.

Some inconsistency exists between Japanse and English corresodence listed above. But, the fact that the interrogative word suggests the category of the answer never changes across over the language. According to this, most systems classify queries. The number of classification is very important and varies from small to large. For example, SQAIQA, NTT communiation Science Laboratories developed, calassifies into 80 categories with hand-made rules[1] and SAIQA-2 into about 160 categories with Support Vector Machines.

It costs too much and takes long time to classify queries into many categories. The following process depends on the result of this classification. So, the error in this step results in the error of the following process and makes the QA system impossible to present the correct answer. From this viewpoint, the precision of classification is very important. However, there must be a certain error with classification with machine learning.

2.1.2 RETRIEVAL of the documents

In this step, documents which may contain the answer from big corpora are retreived according to the query.

Differently from the IR system where query terms are given, terms to retrieve the documents must be selected from the query. In this step, a Japanese full-text search engine like Namazu and techiques for IR are available.

One big problem is the retrieval algorithm. Among the conventional QA systems which use newspaper articles as a knowledge source, some just only use Namazu which adopts simple algorithm or $tf \cdot idf$ algorithm, others establish the original search engines which employ $tf \cdot idf$ algorithm, $Okapi$ algorithm and so on.

Another problem in this step is which phonemes, terms or compound terms should be used as the index of the corpora and query terms. Briefly speaking, a full-text search engine is realized in the way shown below.

1. The index file contains the correspondence between the document and the phonems, terms or compound terms in it.

2. The search engine calculates the similarity between the query terms set and the index of all documents by checking wheter the document contains the query term or not, and then, outputs the documents with high simirality.

So, the form of the terms from phonemes to compound words must be unified between the index and the query terms. Consistency of the dictionary for morphological analysis must be very important.

The system established by Takaki et. al uses the junction of the phonemes as query terms[2].

2.1.3 EXTRACTION of the answer candidates

In this step, answer candidates are extracted from the retrieved documents. What is called "Answer candidates" are the expressions which belong to the category detremined at the ANALYSIS phase.

For example, when the the query ask the name of person by using "Who", expressions which may mean the name of person are extracted in this step. This kind of procedure has been investigated in recent years and implemented as Named Entity Extraction. $NExT$[3] is well-known as the named entity extraction tool. $NExT$ extracts 7 kinds of named entities shown below from raw texts.

7 named entities of $NExT$

ORGANIZATION, PERSON, LOCATION,
DATE, TIME, MONEY, PERCENT

Using the named entity extraction tool, the answer candidates for the query which suggest that the answer is the specific named entity can be extracted. However, the answer candidates for the query which doesn't suggests the category of the named entity extraction tool can't be extracted. The query "What does DVD stand for?" is the proper example for this case. In this case, compounds words and unknown words are extracted as answer candidates in most systems.

The system of Nomoto define 29 named entity categories and establish their original named entity ex-

Table 2. Correspondence between interrogative words and categories of the answer

Interrogative words		Categories
誰，だれ (Who)	→	Person
何処，何所 (where)	→	Place
どこ (where, which)	→	Location, Company, Organization
いつ (When)	→	Time, Date

traction tool[4]. Their named entity extraction tool is based on 178 hand-made rules.

2.1.4 RANKING the candidates

In this step, answer candidates are ranked from the viewpoint of suitability as the answer to the query. Probably there exist plural answer candidates. So, this procedure is essential.

However, you can easily understand that it is very diffcault for the computer to distinguish the correct answer from the other answer candidates which belong to the same category as the correct answer. So, this step is the most difficult procedure in QA systems.

A lot of approaches are proposed for this step. Conventional systems ranked the answer candidates according to the simple distance between the answer candidates and the query terms in the retrieved documents. This approach is based on the assumption that the answer appears in the corpora near the terms in the query. This assumption is appropriate. But no conventional system perform very well on this assumption.

2.2 Problems in the conventional approaches

We will briefly point out the problems in the conventional approaches.

2.2.1 The number of classification

It costs too much time and money to classifiy queries into many categories. Additionaly, if the cllasification is based on the method with machine learning, the accuracy of the classification can't be 100%. However, the accuracy of the classification is very important.

If many categories are defined for the answer expression, named entity extraction tools which deal with many categories are necessary to extract the answer candidates. The F-measure of $NExT$ is about 75 for 7 categories. Defining more categories deterioretes the accuracy of named entity extraction without doubt and as a result the performance of the system will get worse.

Judging from these discussion, small number of categories and good accuracy of classification method are ideal.

2.2.2 The algorithm of RETRIEVAL

In infiormation retrieval like Google and a full-text seach engine like namazu, every query teams are assumed to be equal. However, there are difference between terms in the query in the respect that the documents must contain the term or not. So, dealing with query terms separetely and properly according to parts of speech or other measure is necessary.

Relatively with it, which kind of parts of sppech are used as query terms and the index for the document is also important. We can't deal with it properly with namazu because it works as a black box.

2.2.3 The simple distance between the answer candidate and the terms in the query

The assumption, "the answer appears in the corpora near the terms in the query" is reasonable. Conventional systems use simple distance between them. So, the distance between terms which have a close relation can be bigger than little or no relation. You can see the example for this below.

> Problem of the simple distance
>
> Prince Charles met with earthquake survivors in the flattened city on Bam after talks talks with President Mohammad Khatami earlier Monday in the first visit to Iran by a member of the British royal family in 33 years.

Simple distance in the sentences like this which consist of nested clauses doesn't reflect the relation. Such trends can be seen more clearly in Japanese.

2.3 Necessary techniques

Techniques listed below are necessary to establish Japanese Question Answering system.

- A Japanese morphological analyzer
- A Japanese dependency analyser
- A Japanese full-text search engine
- A Japanese Named Entity Extraction tool

3 Proposed method

In this section, we will explain the proposed method.

3.1 Retrieval Algorithm

At first, we established our original search engine, GBSE, which stands for Geta Based Search Engine. GBSE is based on GETA[1][5]. The chracteristics of GBSE is shown below.

- The response is faster than Namazu.
- Documents are scored according to the tf·idf algorithm.
- Priority can be set to each retrieval terms. The priority consists of two values, high and low. Documents without retrieval terms with high priority can't be retrieved in GBSE.

Our retrieval algorithm using GBSE is as follows.

Indexing:1 Newspaper articles were segmented into paragraphs.

Indexing:2 Each paragraph was morphologically analyzed.

Indexing:3 In the index file, the paragraph id and phonemse in it were recorded. In this phase, phonemes in specific POS were used for the index.

Query:1 The query was also morphologically analyzed and phonemes in specific POS were chosen for the retrieval terms.

Query:2 Phonemes in certain POS such as proper nouns were categorized into essential query terms and the others into optional query terms.

Retrieval:1 The priority of all retrieval words were set to high. GBSE tried to find out paragraphs. If retrieval succeeded, retrieved paragraphs would be used in the next step.

Retrieval:2 If retrieval failed, the priority of the optional query term with the highest TF in all newspaper articles were set to low. Then GBSE tried again. If retrieval failed again, the priority of the optional query terms were set to low one by one according to the TF and GBSE went on trying. In this phase, the priority of the essential retrieval terms were kept high.

Using this algorithm, such benefits shown below are gained.

- Using phoneme in the index and the analyze of the query, dealing with compound words are easy and the consistency of the dictionary is kept.
- It is flexible in which POS are used for the index.
- Tuning of the system is easy because it takes shorter than namazu to index the articles and to retrieve the paragraph.

3.2 Classification of queries according to the expected answer type

Queries are classified into 4 categories in our system. We will explain these categories briefly.

Type 1 Queries which suggested the suffix of the answers were classified into this category. The numbes of the answer candidates with suggested suffix was very small. So, queries in this category were easy to answer for GDQA.

Type1

QAC2-10002-01 Which prefecture does Katakura Kunio, the Japanese ambassador in Iraq at the time of the Gulf War, come from?

湾岸戦争時にイラク大使だった片倉邦雄さんは何県の出身ですか.

Type 2 Queries those interrogative words suggested the type of the answers were classified into this category. For example, if the query had the interrogative words, "WHO", the answer for this query was the name of the person. GDQA extracted the answer candidates for the queries in this category with the named entity extraction tools.

Type2

QAC2-10060-01 Who was the Minister of Finance of the Obuchi Cabinet?

小渕内閣の大蔵大臣は誰ですか.

Type 3 Queries which requested numeric expressions for the answers were classified into this category. Interrogative words and adjuctives, such as "HOW LONG", "HOW TALL" and "HOW MANY", suggested the neumeric expressions for the answers. GDQA extracted the expressions which consisted of numbers and units as the answer candidates fot the queries in this category.

[1] GETA stands for Generic Engine for Transposable Association. It was released from IPA.

Type3

QAC2-10032-01 How tall was Giant Baba?

ジャイアント馬場は身長がどれくらいありますか.

Type 4 Queries which were not classified into categories above were classified into this category. No apparent clues for the answer were found in the query. So, GDQA extracted all nouns and the conjunction of the nouns as answer candidates. As a result, the number of the answer candidates were very big and it got very difficult for GDQA to present the correct answer.

Type4

QAC2-10067 What kind of recycled material was used to make the solar boat Mr. Horie Kennich used to cross the Pacific Ocean alone without making any port calls for first time ever?

世界で初めて太平洋単独無寄港横断に成功した堀江謙一氏のソーラーボートは何を再利用して作られたでしょう.

3.3 Ranking answer candidates with graph structure

Answer candidates are ranked on the assumption that the terms in the query and the answer for it appear near in the retrieved texts.

3.3.1 Construction of the graph structure from retrieved documents

1: Dependency Analysis Each sentence in the retrieved paragraphs was analyzed with CaboCha. The result of the dependency analysis were made up of clauses.

2: Simplification of the result The result of the dependency analysis was directed graph. At first, it was transformed into undirected graph. Then, the clauses were transformed with following rules for simplification.

- Particles and auxiliary verbs were eliminated.
- Punctuation marks and case arcs were eliminated and other symbols were assumed as dependent nouns.
- Unknown words were assumed as independent nouns.
- If a dependent verb or an adjective were contained, the root form of it was made into a stand-alone node.
- If independent nouns was contained, the junction of them was made into a stand-alone node.
- Clauses only with dependent nouns, dependent verbs, dependent adjectives and stopwords were made into dummy nodes. Dummy node only presented the path in the graph structure.

3: Graph Structure One graph structure was made from simplified results with same claused being merged into one node.

4: Shrinking Plural nodes with same meaning existed in the graph structuire. Thesurus were necessary to solve this problem. However, common thesurus that any words cover broad meaning merged node with different meaning in the context. So, the thesurus specialized for the retrieved documents was essential to solve the prpblem. Constructing thesurus was very difficult task. In this phase, we tried to deal with the prpblem partly.

English Abbreviation
JT $\Longleftrightarrow$ 日本たばこ産業

Birth names and common names
イチロー $\Longleftrightarrow$ 鈴木一朗

Name Abbrebiation
小泉首相 $\Longleftrightarrow$ 小泉純一郎首相

3.3.2 Definition of the distance in the graph structure

To define distance between any two nodes in the graph structure, distance between two connected nodes should be defined at first. The distance between two connected node were defined as Equ. (1)

$$distance(node_1, node_2) = \frac{1}{{N_{12}}^2} \quad (1)$$

In Equ. (1), N_{12} meaned the number of links between $node_1$ and $node_2$. The stronger the relation between connected two nodes, the smaller the distance became, and this was a reasonable definition.

After distance between any two connected nodes was defined with Equ. (1), the smallest distance between any two nodes in the graph structure could be calculated with Dijkstra's algorithm.

3.3.3 The score of the answer candidate

According to the assumption, the score of the answer candidates should express the relation with the keywords in the query.

The score of the answer candidates were defined as Equ. (2)

$$Score(Candidate) = \sum_{All\ keywords} Min(A\ keyword, Candidate) \qquad (2)$$

In Equ. (2), Min expressed the smallest distance.

All answer candidates could be sorted according to this score and the system could present the answers in order.

4 Experiments

The experiments on NTCIR-4 QAC2 Task1 and their results are shown in this section.

4.1 Experimental Conditions

Experimental conditions are listed in Tab. 3

Table 3. Experimental conditions

knowledge source	newspaper articles of 4 years

	Mainichi('98, '99)
	# of articles : 220078
	115522('98), 104556('99)
	size : 284MB
	147MB('98), 137MB('99)

	Yomiui('98, '99)
	# of articles : 375980
	132995('98), 242985('99)
	size : 496MB
	184MB('98), 312MB('99)

# of questions	200 NTCIR-4 QAC2 Task1 Formal Run

4.1.1 Maximum number of the paragraphs

The relation between the MRR and the maximum number of the paragraphs used in the phase of answer candidates extraction and the graph structure is shown in the Fig. 2.

This pilot experiments was based on the dataset of the QAC1. This figure suggests that too much paragraphs have no effect on the performance because paragraphs are ordered according to the relation to the query. So, we set the maximum nuber of the paragraphs to 10.

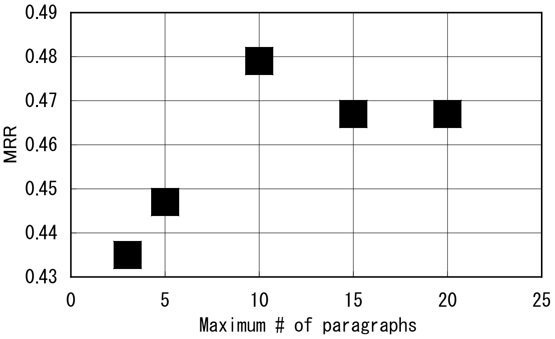

Figure 2. MRR and maximum number of paragraphs

4.2 Evaluation method

System extracts five answer from documents in some order. The inverse of the order of the correct answer, Reciprocal Rank(RR), will be the score of the question. For example, if the second answer is correct, the score will be 1/2. The highest score of the five answers will be the score of the question. If there are several correct answers of a question, system might return one of them, not all of them. Mean Reciprocal Rank(MRR) is used for evaluation of task1. If n set of answers are correct, Mean Reciprocal Rank can be calculated as follows:

$$\begin{aligned} MRR &= \frac{\sum_i^n RR_i}{N} \\ RR_i &= \frac{1}{Rank} \end{aligned}$$

4.3 Results

The system performance is evaluated with the measure, MRR. The result given by the common scoring tool is shown below.

RESULT

Task1 Results
82.8 marks out of 195.0 in TASK1
Average Score:0.425

Qusetion	Answer	Output	Correct
197	385	635	118
Recall	Precision	F-Value	MRR
0.306	0.186	0.231	**0.425**

The MRR measure is very good for camparison between systems. However, we can't understand the performance from it directly. Rate.1st, the rate the system presented the correct answer with 1st rank, was **0.349**. Rate.5, the rate the correct answer was found in the 5 answers of the system's output, was **0.538**.

We will discuss the result in the next section.

5 Discussion

The comparison with other systems is beyond the scope of this paper. In this section, we will discuss the result of our system.

5.1 MRR with each query category

MRR on the queries in the each category defined in 3.2 is shown in Tab. 4.

Table 4. MRR with each query category

	Type 1	Type 2	Type 3	Type 4
MRR	0.49	0.50	0.56	0.28

MRR of Type 4 is terribly lower than the others as we expected. In our classification method, no clue for the answer can't be extracted from queries. So, the number of answer candidates becomes big. Additionaly, answer candidates includes meaningless terms. New approach to extract more information from the queries in Type 4 must be established.

5.2 Response Time

The relation between the response time and the maximum number of the paragraphs used in the phase of answer candidates extraction and the graph structure is shown in the Fig. 3.

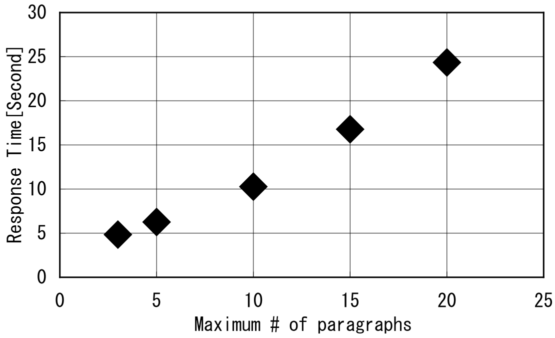

Figure 3. Response time and maximum number of paragraphs

We set the maximun number to 10. So, it takes about 10 seconds to get the answer from GDQA. It's too long to use in social scene. However, in experimental use, it's not too long.

6 Conclusion and future work

6.1 Conclusion

In this paper, we introduced our Question Answering System, GDQA.

Through the experiments, GDQA is still not good enough to be used in the society. However, our new algorithm with graph structure and search engine with GETA took effect.

The performance of the system differs according to the category of the queries.

6.2 Future work

Future work to realize the Question Answering System is listed below.

More information from the queries New techniques to extract more information from the queries, especially from those in the category, Type 4 in our flamework, are necessary.

Thesurus If we can construct the thesurus specialized for the retrieved documents, we can shrink the nodes with same meaning.

Response Time To introduce Question Answering System into the society, quicker response is essential. We must make more efficient algorithm.

References

[1] 佐々木裕, 磯崎秀樹, 平博順, 平尾努, 賀沢秀人, 鈴木潤, 国領弘治, 前田英作. “SAIQA:大量文書に基づく質問応答システム”. 情報学基礎研究会, No. No.064-12, 2001.

[2] Toru TAKAKI, Yoshio ERIGUCHI. “NTT DATA Question-Answering Experiment at the NTCIR-3 QAC”. *Proceedings of the Third NTCIR Workshop*, 2003.

[3] 桝井文人, 鈴木伸哉, 福本淳一. “テキスト処理のための固有表現抽出ツール NExT の開発”. 第8回言語処理学会年次大会発表論文集, pp. 176–179, 3 2002.

[4] Masako NOMOTO, Mithuhiro SATO, Hiroyuki SUZUKI. “NTCIR-3 QAC Experiments at Matsushita”. *Proceedings of the Third NTCIR Workshop*, 2003.

[5] 情報処理技術振興事業協会 (IPA), http://geta.ex.nii.ac.jp/. 汎用連想計算エンジン GETA.